RECURSIVIDAD

Una función es recursiva si se invoca a sí misma.

Cada vez que se invoca a una función se guarda en el stack la dirección de retorno, y se generan las reservas de memoria de las variables locales de la función que se invoca.

Cuando se utilizan funciones recursivas se almacenan en el stack la dirección de retorno y se crean las variables locales de esta, este proceso se repite tantas veces como la función sea invocada.

La recursividad tiene un inconveniente, que si no contiene una condición de salida se generara un algoritmo que no finaliza nunca

Si no contiene una condición de salida, en algún momento, se desbordara el stack.

Produce un error en el funcionamiento del sistema operativo.

Tipos de recursividad

- Recursividad lineal
- Recursividad múltiple

Otra forma de clasificación es

- Recursividad directa
- Recursividad indirecta

Recursividad lineal

Dentro de la función recursiva existe una única invocación a sí misma.

Recursividad múltiple

La función se invoca a sí misma más de una vez dentro de una misma activación.

Recursividad directa

Una función que se llama a sí misma

Recursividad indirecta

Es aquella que se produce cuando se tienen varias funciones que se llaman unas a otras formando un ciclo.